FAST ADVERSARIAL IMPLICIT DIFFUSION FOR MODEL-BASED REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Adversarially guided trajectory diffusion can make model-based reinforcement learning robust by steering a diffusion world model toward low-return trajectories, but this robustness comes at substantial computational cost. We find that diffusion sampling accounts for 80.3% of training wall-clock time. Accelerating this process is nontrivial: shortening the denoising chain changes the allocation of the method's risk budget, while deterministic sampling causes the transition covariance used by adversarial guidance to degenerate. We introduce AID-RRL (Adversarial Implicit Diffusion for Robust Reinforcement Learning), a sampling-time modification that uses a strictly stochastic DDIM-style implicit sampler over a strided sequence of denoising steps and recalibrates the risk budget accordingly. AID-RRL requires no retraining of the diffusion model, return model, or policy. Across nine independently trained MuJoCo Hopper checkpoints, AID-RRL preserves rollout fidelity within a pre-registered equivalence margin while reducing per-batch trajectory-generation cost by 6.0–$12.3\times$ and end-to-end training time by 2.7–$3.1\times$ in single-seed runs. More importantly, a second pre-registered test reveals that neither AID-RRL nor the original full-chain method measurably shifts generated returns toward the intended risk percentile. We show that this failure is structural rather than a consequence of acceleration: the adversarial update scales as $O(\sigma^2)$ while diffusion noise scales as $O(\sigma)$, causing the guidance signal to vanish relative to noise as σ decreases. Thus, AID-RRL both removes a major computational bottleneck and exposes a fundamental limitation of covariance-scaled adversarial diffusion guidance. The same scaling analysis provides a simple diagnostic for evaluating related risk-aware guidance methods before expensive training.

1 INTRODUCTION

Reinforcement learning has produced agents that surpass human-level performance in many domains (Silver et al., 2017; Mnih et al., 2015), yet a policy trained on a clean simulator can fail the moment reality deviates from it: a slightly heavier payload, a sensor with different noise, a joint that does not move quite as modeled. Robustness to this *modeling error* is at least as important as nominal reward in safety-critical domains. Foffano et al. (2026) attain it by combining trajectory-level diffusion, which avoids the compounding error of autoregressive world models (Janner et al., 2022; Rigter et al., 2024), with a Conditional Value at Risk (CVaR) objective, known to confer resilience to model inaccuracy (Osogami, 2012; Chow et al., 2015). Their algorithm, Adversarial Diffusion for Robust Reinforcement Learning (AD-RRL), guides the diffusion model toward the worst α-percentile of returns, and is the method this paper accelerates.

That robustness is not free, and the bill comes due at every single policy-improvement iteration, not once at the end of training. Every synthetic trajectory costs an N-step denoising chain, and each step must also evaluate the policy and differentiate the learned return model Z_{ϕ} for its guidance direction. Because a fresh batch is drawn every iteration of the Dyna loop, sampling dominates wall-clock training time: 80.3% of it, by our own measurement (Section 6.3). The chain runs at sampling time but inside training, so what it costs is a training cost, while the deployed policy runs with no denoiser in the loop at all (Appendix A.2). Reducing the number of denoising steps is therefore the most direct lever on cost, and the one this paper pulls.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Shortening the chain is routine for ordinary diffusion models. It is not routine here, and a naive substitution would fail silently in two specific ways rather than announce itself. First, the guarantee that the induced perturbation ξ stays inside the CVaR risk envelope follows from a per-step budget whose *product* along the chain equals $1/\alpha$; sampling a shorter subsequence with full-chain coefficients mis-spends it, holding the adversary to a budget of 1.46 at $N = 128$, $\tilde{N} = 20$, $\alpha = 0.1$, where recalibration restores the full 10 (the sampler as run in our experiments reaches 9.01; Appendix D). Second, the guidance coefficient is defined through the transition covariance, which degenerates in the deterministic limit, so probability-flow ODE solvers (Lu et al., 2022a; Song et al., 2021b) cannot carry this construction at any order.

We resolve both, and the fix costs nothing at training time. The result is Adversarial Implicit Diffusion for Robust Reinforcement Learning (AID-RRL), a sampling-time modification that reuses the trained diffusion model, return model and policy exactly as they are. Our contributions are threefold:

- **An obstruction, and the diagnostic it yields** (Sections 4.3 and 4.4). Under the AD-RRL calibration, adversarial guidance is undefined whenever the reverse transition is deterministic: the coefficient diverges as κ^{-2} while the displacement vanishes as κ^2, and the failure is silent, ordinary-looking trajectories with the adversary effectively off. The same algebra fixes the adversarial step's magnitude at order σ^2 against noise of order σ, a diagnostic any risk-aware guidance scheme can be checked against on paper.
- **The recalibration that makes acceleration safe** (Section 4.2). The risk-envelope guarantee is chain-length dependent; calibrating the per-step budget to the guided-step count restores the product exactly, where transplanting full-chain coefficients under-spends it by $6.2\times$ at our operating point, a figure following from the chain lengths rather than a measurement. The correction has no detectable effect on any distributional endpoint we measure at $\alpha = 0.1$; Section 6.2 says what that does and does not license.
- **Validation, and a second finding the controls turn up.** The accelerated chain reproduces the full chain's guidance profile and rollout fidelity across nine checkpoints under a pre-registered equivalence margin, cutting per-batch imagination cost 6.0–$12.3\times$ and training end to end 2.7–$3.1\times$ (Section 6). Four conventionally reported statistics do not move when the adversary is switched off, and a pre-registered percentile test finds neither chain steers the return distribution at sampler level, so the diagnostic above is measured rather than merely derived. Training with the sampler in the loop reaches parity at 57% of the steps and a fifth of the GPU-hours before destabilizing on one environment (Section 6.4), traced to an online-tuned policy-guidance scale rather than sampling fidelity.

2 Related Work

Accelerating diffusion sampling. *Implicit samplers* (Song et al., 2021a) exploit the fact that the training objective constrains only the forward marginals, so a trained model can be sampled along a strided subsequence without retraining. *Distillation* (Salimans & Ho, 2022; Song et al., 2023) amortizes the chain into a student network, at the price of a training stage per teacher. *Deterministic high-order solvers* (Lu et al., 2022a; Zhao et al., 2023) integrate the probability-flow ODE (Song et al., 2021b) with large steps and are typically strongest in unguided generation, while *stochastic high-order solvers* (Lu et al., 2022b; **?**; **?**; Karras et al., 2022) integrate the reverse SDE and retain a non-degenerate transition covariance. Our setting reorders this ranking (Table 2): a Dyna loop re-estimates the world model continuously, so a distilled sampler would have to be re-distilled alongside it, and under the AD-RRL calibration adversarial guidance is undefined whenever the reverse transition is deterministic (Section 4.3), removing the probability-flow family. Stochastic high-order solvers are excluded by neither constraint and are a legitimate alternative to the implicit sampler we adopt; we do not evaluate them, and record that as a gap. Training-free guidance work tunes the guidance step size against the noise schedule empirically (Chung et al., 2023; Shen et al., 2024); here the scale is fixed by a risk constraint, sensitive to step count. To our knowledge the interaction between sampler acceleration and a risk-envelope guarantee has not previously been examined.

Diffusion models in RL. Diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020) have modeled dynamics over states (Ajay et al., 2022), actions (Chi et al., 2023) or both (Janner et al., 2022), and policies (Hansen-Estruch et al., 2023) and values (Mazoure et al., 2023), mostly offline;

PolyGRAD (Rigter et al., 2024) is the online, Dyna-style exception and unguided precursor of the method we accelerate. Inference cost has been attacked at the level of the policy (Yu et al., 2025), and risk sensitivity introduced through distributional value guidance (Gong, 2026), but not at the level of a risk-constrained guided sampler, where step count is entangled with the guarantee itself. Guidance (Dhariwal & Nichol, 2021; Ho & Salimans, 2022) conventionally biases generation toward *high*-reward trajectories; AD-RRL (Foffano et al., 2026) uses it in reverse, making the coefficients risk-constrained and hence sensitive to chain length.

Robust and risk-sensitive RL. One line regularizes transition probabilities over an uncertainty set (Derman et al., 2021; Liu et al., 2022); a second frames robustness as a two-player game (Morimoto & Doya, 2005) with an adversary disturbing dynamics (Pinto et al., 2017), actions (Tessler et al., 2019) or states (Stanton et al., 2021); a third optimizes risk measures directly, most often CVaR (Rockafellar & Uryasev, 2000; Tamar et al., 2015; Chow et al., 2015), and domain randomization (Tobin et al., 2017) instead averages over environment parameters. AD-RRL belongs to the third family, realizing the CVaR dual as an adversarial perturbation budgeted over the whole chain, the budget our recalibration must conserve.

3 Background

Robustness through CVaR. An agent acts in a Markov decision process and collects a trajectory $\boldsymbol{\tau}$; write $Z(\boldsymbol{\tau})$ for its discounted return, a random variable because both policy and dynamics are stochastic. We work in a model-based setting: a diffusion model $p_{\boldsymbol{\theta}}$ learns to generate trajectories, and model and policy are improved in alternation following the Dyna paradigm (Sutton, 1991). Risk-averse reinforcement learning maximizes, in place of $\mathbb{E}[Z]$, the *Conditional Value-at-Risk* at level α: the average return over only the worst α fraction of outcomes. The dual form (Artzner et al., 1999), $\mathrm{CVaR}^{\pi}_{\alpha}(Z) = \min_{\xi \in \mathcal{U}^{\alpha,\pi}} \mathbb{E}_{\boldsymbol{\tau}\sim p^{\pi}}[\xi(\boldsymbol{\tau})Z(\boldsymbol{\tau})]$, connects this to robustness: an adversary picks a reweighting ξ of the trajectory distribution to make the expected return as bad as possible, subject to a risk envelope $\mathcal{U}^{\alpha,\pi}$ requiring ξ non-negative, averaging to one, and never exceeding $1/\alpha$ on any trajectory, so maximizing CVaR is maximizing return against a bounded adversarial perturbation of the dynamics (Chow et al., 2015). One feature of that cap drives everything below: $1/\alpha$ is a *total* budget for the whole trajectory, not an allowance per time step (Chow et al., 2015). Implemented inside a denoising chain, that total is spent as a *product* of per-step factors, so shortening the chain is not a neutral change, and making it one is the first thing our method has to do.

Adversarially guided diffusion. A diffusion model generates a trajectory from pure noise by applying a learned denoising step many times over. Each step is Gaussian: the model predicts a mean $\mu_{\boldsymbol{\theta}}(\boldsymbol{\tau}_i, i)$ and the noise schedule fixes a covariance factor $\boldsymbol{\Sigma}_i$. *Guidance* (Dhariwal & Nichol, 2021) steers that process by nudging the mean of every step in a direction supplied by a gradient. AD-RRL (Foffano et al., 2026) points the nudge in the adversarial direction, so that the trajectories generated are the low-return ones the CVaR objective cares about, and each denoising step becomes

$$p_{\boldsymbol{\theta}}(\boldsymbol{\tau}_{i-1} \mid \boldsymbol{\tau}_i) = \mathcal{N}\big(\mu_{\boldsymbol{\theta}}(\boldsymbol{\tau}_i, i) - c_i\, \boldsymbol{\Sigma}_i \boldsymbol{G}_i \boldsymbol{\Sigma}_i,\ \boldsymbol{\Sigma}_i\big), \tag{1}$$

where $\boldsymbol{G}_i$ is the gradient of a learned return model, pointing toward higher return with the minus sign turning the step against it, and the scalar c_i sets how hard to push. Trajectories here are matrix-valued rather than vector-valued, so the reverse transition is matrix-normal and carries a covariance factor on each side of $\boldsymbol{G}_i$; the normalizing quadratic form $\mathrm{tr}(\boldsymbol{G}_i^T \boldsymbol{\Sigma}_i \boldsymbol{G}_i \boldsymbol{\Sigma}_i)$ is quartic, not quadratic, in the entrywise noise scale, where the $\kappa^{\pm 2}$ exponents of Section 4.3 originate (Appendix A.3).

Everything our acceleration must preserve is contained in how the c_i are chosen. Foffano et al. (2026) show that running the whole chain reweights the generated distribution by some factor ξ, and that keeping ξ inside the risk envelope reduces to keeping it below $1/\alpha$. They meet that bound by splitting $1/\alpha$ across the guided steps *multiplicatively*: each step i is allotted a factor $\eta_i \geq 1$ with $\prod_i \eta_i(\alpha, N) = 1/\alpha$, and the guidance strength spending exactly the allotted factor is

$$c_i = \sqrt{2 \log \eta_i(\alpha, N) \big/ \mathrm{tr}(\boldsymbol{G}_i^T \boldsymbol{\Sigma}_i \boldsymbol{G}_i \boldsymbol{\Sigma}_i)}. \tag{2}$$

The standard choice gives every step the same share, $\eta_i(\alpha, N) = (1/\alpha)^{1/N}$, under two mild conditions (bounded trajectories, diagonal transition covariance with entries below one) that the cosine

schedule (Nichol & Dhariwal, 2021) used by the world model satisfies; we take these as given and refer to Foffano et al. (2026) for the derivation. In (2) the allotted budget enters through a logarithm in the numerator, the noise level through the denominator; the first matters in Section 4.2, the second in Section 4.3 and again in Section 6.2.

4 Adversarial Implicit Sampling

A diffusion model trained with N denoising steps can be sampled with far fewer, by visiting only a subsequence of them, standard and requiring no retraining (Song et al., 2021a). Doing so to an *adversarially guided* model breaks the risk guarantee, because that guarantee is a budget split across however many steps the chain has, so a shorter chain must give each step a larger share; and the short chain must keep injecting noise at every step, because the guidance strength divides by the noise level and a noiseless step divides by zero. This section makes those claims precise.

4.1 Implicit sampling on a subsequence

A diffusion model is trained on a criterion constraining only how noisy a trajectory looks at each level of corruption, not the order in which the noise was added, and Song et al. (2021a) observed that many generative processes are consistent with that same criterion, so a model trained once can be sampled by any of them, in particular by one that skips steps. Choose a subsequence $\mathcal{S} = \{s_1 < \cdots < s_{\tilde{N}}\}$ of the N training steps, start from pure noise at the largest retained step, and denoise along $\mathcal{S}$ alone. Each step remains Gaussian, with a mean supplied by the same trained network and covariance factors

$$\boldsymbol{\Sigma}^{\mathcal{S}}_{s_j} = \sigma^2_{s_j} \boldsymbol{I}, \qquad \sigma_{s_j} = \kappa\, \rho_j, \tag{3}$$

where ρ_j is fixed by the noise schedule and the pair of steps being bridged, and $\kappa \in [0, 1]$ controls how much noise the reverse process injects (the parameter written η by Song et al. (2021a), renamed because η already denotes the per-step budget shares). Setting $\kappa = 1$ reproduces the usual ancestral variance; $\kappa = 0$ makes the sampler deterministic and is the setting most acceleration work prefers. Appendix B gives ρ_j and the mean in full.

Two consequences follow. First, generating a trajectory now costs $\tilde{N}$ steps instead of N, chosen at sampling rather than training time. Second, and this is what makes the adversarial case work at all, the step is still a Gaussian described by a mean and a covariance, so any construction written purely in those two objects carries over unchanged, with i replaced by s_j and N by $\tilde{N}$. The guided step (1) is such a construction, so it transfers verbatim, reusing the same trained network, return model and policy. What does *not* transfer is the calibration of c_{s_j}, and what fails outright is the deterministic setting $\kappa = 0$.

4.2 Recalibrating the risk budget

The risk budget is a product: each guided step contributes a factor, and the factors must multiply to $1/\alpha$. Take coefficients calibrated for a 128-step chain, run them on a 22-step chain, and only 22 of the 128 factors ever get applied. The product falls far short of $1/\alpha$, and the adversary is quietly held to a fraction of the risk level the method advertises. The fix is to give each retained step a correspondingly larger share, and the denominator to use is the number of steps that actually carry guidance.

Concretely, let $\mathcal{S}$ have realized length $\tilde{N}$ and let n of its steps carry guidance. Coefficients calibrated for the full chain, $\eta_i(\alpha, N) = (1/\alpha)^{1/N}$, enforce a budget of only $(1/\alpha)^{n/N}$, since only n of the N factors are ever applied. Recalibrating to the guided-step count,

$$\eta_j(\alpha, n) = \left(\tfrac{1}{\alpha}\right)^{1/n}, \quad c_{s_j} = \sqrt{2 \log \eta_j(\alpha, n) \big/ \operatorname{tr}(\boldsymbol{G}^T_{s_j} \boldsymbol{\Sigma}^{\mathcal{S}}_{s_j} \boldsymbol{G}_{s_j} \boldsymbol{\Sigma}^{\mathcal{S}}_{s_j})}, \tag{4}$$

restores the product to $1/\alpha$ exactly, and the two conditions behind (2) continue to hold under the implicit covariance (3), so the derivation of Foffano et al. (2026) goes through unchanged with $i \mapsto s_j$ and $N \mapsto n$. Appendix C gives the short argument. The sampler used in Section 6 calibrates against $\tilde{N}$ rather than $n = \tilde{N} - 1$, so its budget product is 0.90, 0.87 and 0.81 of $1/\alpha$ rather than

one; the correction rescales each coefficient by at most 1.05 (Appendix D), and the $6.2\times$ gap quoted throughout is between this as-run sampler and the uncalibrated one.

4.3 Why the sampler must remain stochastic

The obvious way to accelerate further is to remove the noise from the reverse process and solve a deterministic equation. That option is closed here, and the reason is visible in (4): the guidance strength divides by the noise variance, so shrinking the noise inflates the coefficient while the displacement it produces shrinks faster, and the adversary disappears exactly when its nominal strength is largest.

Under the calibration (2), as $\kappa \to 0$ the coefficient c_{s_j} diverges as κ^{-2} while the displacement it applies to the mean vanishes as κ^2, even though the budget it is calibrated to spend does not. Adversarial guidance so calibrated is therefore undefined for any deterministic reverse transition. Both exponents follow from the size of the step the guidance actually applies,

$$\left\| c_{s_j} \boldsymbol{\Sigma}_{s_j}^{\mathcal{S}} \boldsymbol{G} \boldsymbol{\Sigma}_{s_j}^{\mathcal{S}} \right\|_F = \sqrt{2 \log \eta_j}\, \sigma_{s_j}^2, \tag{5}$$

together with $\sigma_{s_j} \propto \kappa$ from (3); Appendix C has the derivation. Equation (5) turns the obstruction from a limit statement into a pair of exponents that can be measured, and Section 6.1 measures them.

Any guidance whose strength is calibrated against a per-step noise budget needs that budget nonzero, so the whole family of accelerations built on the probability-flow ODE (Song et al., 2021b; Lu et al., 2022a; Zhao et al., 2023) is unavailable at any solver order. The exclusion is scoped to the calibration (2), not to adversarial guidance in general: a scheme budgeting the perturbation in trajectory norm, or against the predicted clean trajectory, would have no such singularity and could in principle be carried by a deterministic solver, which Appendix A.5 argues is the more valuable next step. Higher-order stochastic solvers remain available though we do not test them. We set $\kappa = 1$ throughout.

4.4 A fixed-norm adversarial step

Equation (5) says something stronger than the deterministic-limit argument needs, and it is the fact the experiments of Section 6.2 turn on, so we state and isolate it here rather than leave it as a byproduct of that argument.

The size of the adversarial step does not depend on the return model. For every step, $\left\| c_{s_j} \boldsymbol{\Sigma}_{s_j}^{\mathcal{S}} \boldsymbol{G} \boldsymbol{\Sigma}_{s_j}^{\mathcal{S}} \right\|_F = \sqrt{2 \log \eta_j}\, \sigma_{s_j}^2$, in which $\boldsymbol{G}$ does not appear: the gradient term in the denominator of (4) exactly cancels the one in the numerator. The learned return model chooses the *direction* the adversary pushes; the *distance* it pushes is fixed by the risk budget and the noise schedule alone.

This is a property of the construction we inherit, not of our acceleration, and to our knowledge it has not been stated before. The displacement is calibrated against the noise *variance* $\sigma_{s_j}^2$ while the noise it competes with scales as σ_{s_j}, which Section 6.2 shows is why the risk mechanism does not do what it is supposed to; and because no return model changes the magnitude, that finding cannot be an artifact of the particular $Z_{\boldsymbol{\phi}}$ we measured it on. One consequence bounds a question the experiments raise before they are run: since the correction of Section 4.2 only rescales c_{s_j} by $\sqrt{\tilde{N}/n}$, the uncorrected numbers of Section 6 cannot understate the risk budget by more than that known factor, well inside batch-to-batch noise; Appendix D derives and tabulates the bound.

4.5 The sampler

The surrounding algorithm is that of AD-RRL and is unchanged; AID-RRL differs only in the trajectory-sampling subroutine (Algorithm 1), stated for a generic subsequence $\mathcal{S}$ and noise scale κ so one implementation covers both methods. At every visited step we inpaint a real starting state $\boldsymbol{s}_0$, evaluate $\boldsymbol{G}_{s_j} = \nabla_{\boldsymbol{s}} Z_{\boldsymbol{\phi}}$, and form the coefficient from (4). The gradient is with respect to $\boldsymbol{s}$ alone, so only states are adversarially corrupted; actions stay consistent via the PolyGRAD guidance mechanism (Rigter et al., 2024). Training is untouched, which is the point of the construction: the model remains a noise-prediction network fit by the usual objective with i drawn uniformly over $\{1, \ldots, N\}$, and $\mathcal{S}$ is a decision made at sampling time, so switching between AD-RRL and

Algorithm 1 Adversarial Implicit Trajectory Sampling. Setting $\mathcal{S} = \{0, \dots, N\}$, $\kappa = 1$ recovers the ancestral sampler of AD-RRL (Foffano et al., 2026); a strided $\mathcal{S}$ with $\tilde{N} \ll N$ gives AID-RRL.

Input: adversarial denoising model $\bar{p}_{\boldsymbol{\theta}}$; reward model $Z_{\boldsymbol{\phi}}$; buffer $\mathcal{D}$; level α; subsequence $\mathcal{S} = \{0 = s_1 < \cdots < s_{\tilde{N}}\}$; noise scale $\kappa > 0$; guided-step count $n = \tilde{N} - 1$
$\hat{\boldsymbol{\tau}}_{s_{\tilde{N}}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$; $\quad \boldsymbol{s}_0 \sim \mathcal{D}$
for $j = \tilde{N}, \dots, 2$ **do**
  set $\hat{\boldsymbol{s}}_0 \leftarrow \boldsymbol{s}_0$ in $\hat{\boldsymbol{\tau}}_{s_j}$
  $\boldsymbol{\Sigma}^{\mathcal{S}}_{s_j} = \sigma^2_{s_j} \boldsymbol{I}$ with σ_{s_j} from (3); $\quad c_{s_j}$ from (4) with share $\eta_j(\alpha, n)$
  $\hat{\boldsymbol{\tau}}_{s_{j-1}} \sim \mathcal{N}\big(\mu^{\mathcal{S}}_{\boldsymbol{\theta}}(\hat{\boldsymbol{\tau}}_{s_j}, s_j) - c_{s_j} \boldsymbol{\Sigma}^{\mathcal{S}}_{s_j} \boldsymbol{G}_{s_j} \boldsymbol{\Sigma}^{\mathcal{S}}_{s_j},\ \boldsymbol{\Sigma}^{\mathcal{S}}_{s_j}\big)$
end for
return $\hat{\boldsymbol{\tau}}_0$

AID-RRL needs no retraining and no change to any learned parameter. Subsequence *spacing* is unexplored: we use uniform stride throughout, although Section 6.2 shows the adversarial displacement concentrated in the noisiest steps, arguing for a non-uniform schedule.

5 METHODS

We ask five questions: whether the risk budget survives acceleration, whether the deterministic limit is usable, where the guided trajectories land, what the acceleration costs in fidelity and saves in compute, and whether a policy trained with it behaves the same. This section fixes the setup, the endpoints and the pre-registered criteria for each; Section 6 reports what was found, in the same order.

5.1 COMMON SETUP

All sampler-level experiments are on MuJoCo (Todorov et al., 2012; Brockman et al., 2016) at $\alpha = 0.1$, $N = 128$, $\kappa = 1$, comparing the ancestral sampler against implicit samplers with $\tilde{N} \in \{10, 15, 20\}$ (stride 22, 16, 11 realized steps). Measurements hold a trained agent fixed and vary only the sampler, five batches per configuration unless stated. Hardware, software and seeding are in Appendix A.4; every comparison was timed on the same machine within a single job.

5.2 BUDGET CONSERVATION AND THE DETERMINISTIC LIMIT

Two controls test whether an endpoint responds to the guidance at all. The *uncalibrated* arm runs the implicit chain with full-chain coefficients, so its enforced budget is wrong by the $6.2\times$ of Section 4.2; the $\kappa = 10^{-2}$ arm shrinks the injected noise so that, by (5), the adversary is switched off to within a factor of 13 000 while the interface is unchanged. Under both we record mean imagined return, dynamics MSE, reward MSE, terminal fraction and the summed adversarial displacement, and ask which of them move. The deterministic limit is probed by sweeping κ from 1 to 10^{-2} at $\tilde{N} = 20$ and fitting log-log slopes of c and of the displacement against the exponents -2 and $+2$ that Section 4.3 predicts. The budget product $\alpha \prod_j \eta_j$ is algebraically determined and is reported as an implementation check in Appendix D rather than as a result.

5.3 WHERE THE GUIDED TRAJECTORIES LAND

Whether spending the budget puts the generated trajectories where CVaR says they belong is a separate question from whether it is spent, and one we settled in advance rather than after the fact. From one batch of conditioning states we draw 512 trajectories under each of four conditions: *unguided* (the adversarial displacement zeroed, transition covariance, injected noise and policy guidance untouched), ancestral, implicit ($\tilde{N} = 20$), and uncalibrated implicit. The unguided arm supplies the reference distribution and the endpoint is the quantile q of each guided arm's mean return within it; reaching the $\alpha = 0.1$ percentile is the claim under test, against a falsifier fixed before the data were drawn ($q > 0.35$). Two further comparisons were pre-registered on the same data: the implicit arm

against the ancestral arm, with a Δq threshold of 0.15, and the uncalibrated arm against the calibrated one. This experiment is fully seeded and reproduces bit-for-bit on every non-timing quantity (Appendix A.4).

5.4 Acceleration costs and benefits

We evaluate both samplers against trained Hopper agents, recording wall-clock per imagined batch, the open-loop dynamics error obtained by replaying each imagined action sequence in the simulator from the matched initial state, the first-step reward error, and the mean imagined return, with the sampler as the only varying factor. The per-batch measurement is taken first on one reference checkpoint (1.5M steps) and then repeated on eight further independently trained Hopper checkpoints. Fidelity is judged by an equivalence test on the terminal dynamics-MSE ratio against a margin of 1.10, fixed before any of these data were collected (Schuirmann, 1987; Lakens, 2017). The end-to-end saving is predicted from the measured imagination share of training wall-clock by Amdahl's law (Amdahl, 1967) and checked against two independent routes: a decomposition of component times, and the training runs of Section 5.5.

5.5 Training with the sampler in the loop

The measurements above hold a trained agent fixed. Whether a policy trained *with* the implicit sampler in the loop reaches the same behavior is a separate question, because fidelity loss compounds across policy updates. We train a second agent from scratch on the same seed with $\tilde{N} = 20$ in the loop, identical in every other argument, and run both arms to 1.5M environment steps. The pre-registered test is on *final* return against the band $[2989, 3617]$. To diagnose any divergence, all 128 logged metrics are screened for movement in the 70k steps before onset, and test-time robustness is compared descriptively over a mass/gravity/friction sweep of 18 conditions (Appendix F). A second paired run on Walker2d-v3 and a $K = 1$ replication matrix across Hopper-v3, Walker2d-v3 and HalfCheetah-v3 test whether the outcome is a property of the sampler.

6 Results

6.1 Budget conservation

Most candidate endpoints here cannot serve as evidence, either because they are algebraically determined rather than measured (the budget product, the $6.2\times$ gap between arms; Appendix D) or because they do not move when the mechanism they describe is switched off: mean imagined return, dynamics MSE, reward MSE and terminal fraction are all flat under the uncalibrated arm, whose budget is wrong by $6.2\times$, and under $\kappa = 10^{-2}$, which disables the adversary to within a factor of $13\,000$. This is why the fidelity results of Section 6.3 are claims about the *world model* and never the guidance, and what motivated the percentile endpoint of Section 6.2. The summed adversarial displacement is the exception, falling $2.4\times$ under the controls with the two arms in agreement (0.266 over 127 steps versus 0.274 over 21; Figure 1b). The κ sweep confirms the exponents of Section 4.3 to two decimals (-2.00, $+2.00$): a deterministic solver disables the adversary silently, generating unguided samples under a guided interface. Appendix A.6 gives all of this in full.

6.2 Where the guided trajectories land

Section 6.1 establishes that the budget is spent and that it must be. Whether spending it steers the generated distribution is what the percentile endpoint of Section 5.3 measures. Reaching the $\alpha = 0.1$ percentile would require the mean to fall by 8.95; the ancestral chain moves it by $+0.0025$, which is 0.03% of that, and upward. Against the pre-registered falsifier the percentile claim is **refuted**, for the full chain as squarely as for the accelerated one (Table 3 and Figure 4). The cause is structural. Equation (5) fixes the adversarial step at $\sqrt{2 \log \eta_j}\, \sigma_{s_j}^2$ against injected noise of norm $\sigma_{s_j} \sqrt{d}$, with $\sigma_{s_j} < 1$ throughout the schedule, so an $O(\sigma^2)$ displacement is dominated at every step by an $O(\sigma)$ one: it never exceeds 1.4% of the noise beside it (Figure 3). This comes from (2), not from anything acceleration does, which is why the ancestral chain fails the test as squarely as the implicit one; and because (5) makes the magnitude independent of $\boldsymbol{G}$, any risk-aware guidance scheme can run the same comparison of step scale to noise scale on paper before committing a training run.

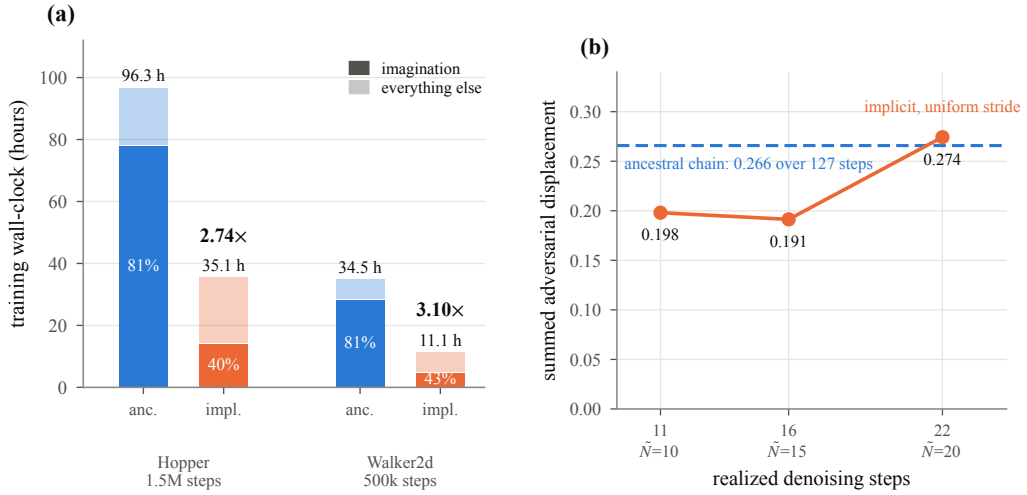


Figure 1: Faster, and the guidance profile is conserved. **(a) Imagination dominates training time.** End-to-end wall-clock for matched ancestral and implicit ($\tilde{N} = 20$) runs, split into imagination (solid, share in percent) and everything else (faded); the imagination share is reconstructed from per-call imagination time and the number of policy updates. Imagination is about four fifths of the ancestral run on both environments and 40–43% of the implicit one, which is why a $5.9\times$ per-batch saving becomes 2.7–$3.1\times$ end to end. **(b) The summed adversarial displacement survives acceleration.** Displacement summed over the chain for the three implicit configurations, five batches each on one trained Hopper agent, 95% intervals; the dashed line is the 127-step ancestral chain. The 22-step chain ($\tilde{N} = 20$) matches the ancestral value and the 11-step chain retains about three quarters of it.

Table 1: Sampler cost and fidelity on a trained agent (1.5M steps). "Steps" is the realized subsequence length; "Dyn. MSE" the open-loop error at the final horizon step, mean over five batches. Reward MSE and mean imagined return are in Appendix E; both differ by under 1% across rows and (Section 6.1) do not respond when guidance is disabled, so neither is evidence about the guidance.

	Steps	s/batch	Speedup	Dyn. MSE
Ancestral (N=128)	128	0.8521	1.00×	0.1230
Implicit ($\tilde{N}$=20)	22	0.1437	5.93×	0.1241
Implicit ($\tilde{N}$=15)	16	0.1032	8.26×	0.1286
Implicit ($\tilde{N}$=10)	11	0.0699	12.19×	0.1283

This is one sampler-level batch against guidance that acts cumulatively over 375 000 policy updates, so it neither shows that adversarial diffusion fails end to end nor addresses the robustness results of Foffano et al. (2026); it bounds what any sampler-level experiment can establish about the risk mechanism. Appendix A.6 gives the four-arm table and the two further pre-registered comparisons.

6.3 Acceleration Costs and Benefits

On the single reference checkpoint (Table 1) the cost falls $5.93\times$, $8.26\times$ and $12.19\times$ at $\tilde{N} = 20, 15, 10$, tracking the reduction in realized steps almost exactly. Repeating this on the eight further independently trained Hopper checkpoints gives a near-deterministic saving of $6.00 \times [5.94, 6.07]$, $8.31 \times [8.24, 8.38]$ and $12.27 \times [12.23, 12.31]$, a spread of about 1%; these are the figures quoted in the abstract and conclusion. Imagination dominates training wall-clock, 80.3% of it (77.3 of 96.3 hours over 375 000 calls; Figure 1a), which bounds the result: Amdahl's law caps the end-to-end gain at $5.06\times$, and at $\tilde{N} = 20$ the $5.93\times$ per-batch reduction predicts $3.01\times$. Two independent routes agree: decomposing component times gives $3.03\times$, and the training runs of Section 6.4 give $2.74\times$ on Hopper (35.2 against 96.3 hours) and $3.10\times$ on Walker2d (11.1 against 34.5 hours), both

in Figure 1a. The per-batch factor is not the training factor, so $\tilde{N} = 10$'s $12.27\times$ does not become a $12\times$ training speedup, and it retains only 75% of the ancestral displacement against $\tilde{N} = 20$'s 3.2% (Figure 1b); we recommend $\tilde{N} = 20$.

6.4 Training with the sampler in the loop

Up to 860k steps the two arms are hard to tell apart (Figure 2): the implicit arm peaks at 3220 against the ancestral arm's 3212 at the same step, $0.974\times$ its eventual best of 3305, at a fifth of the wall-clock (35.2 against 96.3 hours), and robustness survives too where it can be fairly measured (the mass/gravity/friction sweep gives a median absolute difference of 1.1% across 18 conditions, descriptively rather than inferentially; Figure 6, Appendix F). Then, at 930k steps, the accelerated arm's return falls from 3191 to 56 inside one evaluation window and never recovers (final return 39.1); the ancestral arm's own transient dips each recover and it ends at 3288. The pre-registered test was on *final* return against the band $[2989, 3617]$ (Section 5.5), so on its own terms the claim that training with the implicit sampler matches the ancestral arm is refuted, reported as pre-registered rather than retreating to the peak. Accumulated fidelity loss from the shorter chain is the obvious explanation and the one the data rule out: across 128 logged metrics screened before onset the largest excess on any pre-registered hypothesis is $|z| = 1.20$, and the accelerated arm's imagination error runs 3–$5\times$ *lower* than the baseline's throughout. What differs is a policy-guidance control loop inherited from PolyGRAD, held at $5.1\times$ the baseline's operating point and slaved to its own tracking error once the collapse begins ($\rho = 0.983$ against 0.434); the loop is indistinguishable between arms *before* onset, so this run cannot separate a loop that initiates the collapse from one that only amplifies it. Appendix F gives the full screen, loop analysis, robustness sweep, and the second Walker2d run, which degrades gradually rather than breaking.

7 Limitations

Our claims concern faithful acceleration of AD-RRL's sampler and what sampler-level measurement can and cannot detect. All end-to-end comparisons are against the ancestral sampler, which establishes that acceleration preserves performance but not *why* that performance arises: the mechanism acting cumulatively over 375 000 policy updates and PolyGRAD's world model alone are both consistent with our results (Section 6.2). An unguided arm would separate them, and we make no claim that depends on the answer. The percentile claim does not hold at sampler level for the ancestral method either, so this is a property of the method we accelerate rather than something the acceleration introduces; by the derivation of Section 4.4 we expect it not to be specific to our checkpoint, environment or α, though we have measured only one of each. It also leaves open the end-to-end robustness Foffano et al. (2026) report, which operates at a different level.

8 Conclusion

We introduced AID-RRL, a sampling-time acceleration for adversarially guided diffusion in model-based reinforcement learning. By recalibrating the per-step risk budget to the denoising steps actually taken and retaining stochastic transitions, AID-RRL requires no retraining and preserves the full chain's rollout fidelity across nine independently trained checkpoints. It reduces per-batch trajectory-generation cost by 6.0–$12.3\times$ and end-to-end training time by 2.7–$3.1\times$ in single-seed runs. The experiments also reveal a more fundamental limitation. Neither the accelerated sampler nor the original full chain measurably shifts generated returns toward the intended risk percentile at sampler level. Our analysis shows why: the adversarial displacement scales as $O(\sigma^2)$ while diffusion noise scales as $O(\sigma)$, causing guidance to weaken relative to noise as σ decreases. This scaling provides a simple diagnostic for related risk-aware diffusion methods before expensive training. End-to-end training further shows that sampler fidelity alone does not guarantee system-level stability. The accelerated arm initially reaches comparable performance at substantially lower GPU cost, but later destabilizes on one of two environments in association with a shift in the online-tuned policy-guidance scale. These results separate three questions that should be tested independently: whether the sampler remains faithful, whether guidance produces its intended effect, and whether the surrounding training system remains stable. AID-RRL addresses the first while making the latter two directly testable.

AI USE STATEMENT

Generative AI tools were used during the research process to assist with brainstorming, code development, and manuscript editing. All research questions, methodology, experimental design, analyses, and conclusions were developed and verified by the authors. The authors take full responsibility for the content of the paper.

REPRODUCIBILITY STATEMENT

The hardware, software versions and seeding procedure behind every timing and sampler measurement are stated in Appendix A.4, including which sweeps reproduce bit-for-bit and which redraw conditioning states. The sampler is given in full as Algorithm 1, the implicit reverse step and its schedule factor in Appendix B, and the derivations behind the recalibration and the displacement norm of Section 4 in Appendix C. Appendix D reports the algebraically determined quantities against the values the implementation logs, so a reimplementation can be checked step by step. Appendix E lists the architectures and every hyperparameter, including the reward-model ensemble size that our checkpoints were inherited under. The sampler, the pre-registration documents and the logged metric histories will be released with the paper; links are withheld here because they would identify the authors under the double-blind policy.

REFERENCES

Anurag Ajay, Yilun Du, Abhi Gupta, Joshua Tenenbaum, Tommi Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision-making? *arXiv preprint arXiv:2211.15657*, 2022.

Gene M Amdahl. Validity of the single processor approach to achieving large scale computing capabilities. In *AFIPS Spring Joint Computer Conference*, pp. 483–485, 1967.

Philippe Artzner, Freddy Delbaen, Jean-Marc Eber, and David Heath. Coherent measures of risk. *Mathematical finance*, 9(3):203–228, 1999.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI Gym. *arXiv preprint arXiv:1606.01540*, 2016.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, pp. 02783649241273668, 2023.

Yinlam Chow, Aviv Tamar, Shie Mannor, and Marco Pavone. Risk-sensitive and robust decision-making: a cvar optimization approach. *Advances in neural information processing systems*, 28, 2015.

Hyungjin Chung, Jeongsol Kim, Michael T Mccann, Marc L Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *International Conference on Learning Representations*, 2023.

Esther Derman, Matthieu Geist, and Shie Mannor. Twice regularized mdps and the equivalence between robustness and regularization. *Advances in Neural Information Processing Systems*, 34: 22274–22287, 2021.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Daniele Foffano, Alessio Russo, and Alexandre Proutiere. Adversarial diffusion for robust reinforcement learning. *Advances in Neural Information Processing Systems*, 38:68627–68658, 2026.

Shiqiang Gong. RS-Diffuser: Risk-sensitive diffusion planning with distributional value guidance. *arXiv preprint arXiv:2606.27766*, 2026.

Philippe Hansen-Estruch, Ilya Kostrikov, Michael Janner, Jakub Grudzien Kuba, and Sergey Levine. IDQL: Implicit q-learning as an actor-critic method with diffusion policies. *arXiv preprint arXiv:2304.10573*, 2023.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991*, 2022.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, 2022.

Daniël Lakens. Equivalence tests: A practical primer for t tests, correlations, and meta-analyses. *Social Psychological and Personality Science*, 8(4):355–362, 2017.

Zijian Liu, Qinxun Bai, Jose Blanchet, Perry Dong, Wei Xu, Zhengqing Zhou, and Zhengyuan Zhou. Distributionally robust q-learning. In *International Conference on Machine Learning*, pp. 13623–13643. PMLR, 2022.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-Solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022a.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-Solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv preprint arXiv:2211.01095*, 2022b.

Bogdan Mazoure, Walter Talbott, Miguel Angel Bautista, Devon Hjelm, Alexander Toshev, and Josh Susskind. Value function estimation using conditional diffusion models for control. *arXiv preprint arXiv:2306.07290*, 2023.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Jun Morimoto and Kenji Doya. Robust reinforcement learning. *Neural computation*, 17(2):335–359, 2005.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pp. 8162–8171. PMLR, 2021.

Takayuki Osogami. Robustness and risk-sensitivity in markov decision processes. *Advances in neural information processing systems*, 25, 2012.

Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial reinforcement learning. In *International conference on machine learning*, pp. 2817–2826. PMLR, 2017.

Marc Rigter, Jun Yamada, and Ingmar Posner. World models via policy-guided trajectory diffusion. *Transactions on Machine Learning Research*, 2024. arXiv:2312.08533.

R Tyrrell Rockafellar and Stanislav Uryasev. Optimization of conditional value-at-risk. *Journal of Risk*, 2(3):21–41, 2000.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022.

Donald J Schuirmann. A comparison of the two one-sided tests procedure and the power approach for assessing the equivalence of average bioavailability. *Journal of Pharmacokinetics and Biopharmaceutics*, 15(6):657–680, 1987.

Yifei Shen, Xinyang Jiang, Yezhen Wang, Yifan Yang, Dongqi Han, and Dongsheng Li. Understanding and improving training-free loss-based diffusion guidance. In *Advances in Neural Information Processing Systems*, 2024. arXiv:2403.12404.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*, 2017.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. PMLR, 2015.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021b.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning*, 2023.

Samuel Stanton, Rasool Fakoor, Jonas Mueller, Andrew Gordon Wilson, and Alex Smola. Robust reinforcement learning for shifting dynamics during deployment. 2021.

Richard S Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *ACM Sigart Bulletin*, 2(4):160–163, 1991.

Aviv Tamar, Yonatan Glassner, and Shie Mannor. Optimizing the CVaR via sampling. In *AAAI Conference on Artificial Intelligence*, 2015.

Chen Tessler, Yonathan Efroni, and Shie Mannor. Action robust reinforcement learning and applications in continuous control. In *International Conference on Machine Learning*, pp. 6215–6224. PMLR, 2019.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 23–30. IEEE, 2017.

Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.

Shu-Ang Yu, Feng Gao, Yi Wu, Chao Yu, and Yu Wang. D3P: Dynamic denoising diffusion policy via reinforcement learning. *arXiv preprint arXiv:2508.06804*, 2025.

Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. UniPC: A unified predictor-corrector framework for fast sampling of diffusion models. In *Advances in Neural Information Processing Systems*, 2023.

A Additional material

This appendix expands on points the main text states briefly.

A.1 Admissibility of the acceleration families

The main text takes for granted that only two of the five acceleration families are available in this setting. Table 2 records why.

Table 2: Admissibility of each acceleration family under the AD-RRL calibration (2). "Survives re-fit" asks whether the accelerator remains valid when the Dyna loop re-estimates the world model. "Stoch." asks whether the reverse transition retains a non-degenerate covariance, which Section 4.3 shows is necessary for adversarial guidance to be defined. The last column records admissibility, not measured performance: the final row is admissible and untested here.

Family	Training-free	Survives re-fit	Stoch.	Admissible here
Implicit, strided	✓	✓	✓	✓(used)
Distillation	–	–	✓	–
Probability-flow ODE	✓	✓	–	–
Consistency / flow-map	–	–	–	–
Stochastic solver	✓	✓	✓	✓(untested)

A.2 Three clocks

Three processes run in an adversarially guided Dyna agent, on three different clocks, and the method of Section 4 touches exactly one of them. Distinguishing them resolves an ambiguity of vocabulary: in the diffusion literature *sampling time* names the moment a trained generative model is run, while in reinforcement learning *test time* names deployment. In a Dyna agent these are not the same moment, and the distance between them is where the cost of this method sits.

The first clock fits the diffusion model: trajectories collected from the environment are corrupted by the forward noising process and the network $\boldsymbol{\theta}$ is trained to invert one such corruption. It is the only clock on which weights change. The second is the policy-improvement loop, in which every iteration calls the trained model to imagine a fresh batch and every trajectory is one complete reverse chain of N guided denoising steps, each of which also differentiates $Z_{\boldsymbol{\phi}}$. That is where the 80.3% of wall-clock reported in Section 6.3 is spent, and it is the only clock this paper modifies. The third is deployment, on which no denoising step is evaluated and the diffusion model can be discarded entirely.

Two consequences follow. The modification we present is a sampling-time one in the diffusion sense, but the saving it produces is a *training* saving, because the second clock lies inside training rather than after it; the accelerated sampler is absent from deployment and every cost claim in this paper is a claim about that clock. And the first clock is not entered once: a Dyna agent re-estimates its world model as new data arrives, so any acceleration requiring a training stage of its own, distillation into a student most prominently, would have to be repeated each time the teacher moves. That is the constraint recorded in Table 2.

A.3 Matrix-valued trajectories and the two covariance factors

This point is easy to pass over and it decides the exponents in Section 4.3, so we state it in full here. A trajectory occupies a horizon-by-feature array rather than a vector, and the reverse transition of AD-RRL is matrix-normal: it carries a left covariance over time steps and a right covariance over features, both equal to $\boldsymbol{\Sigma}_i$ in the isotropic case we work in. The guidance drift is accordingly $\boldsymbol{\Sigma}_i \boldsymbol{G}_i \boldsymbol{\Sigma}_i$ with $\boldsymbol{G}_i$ the return-model gradient arranged in the same array shape, and the quadratic form that normalizes the coefficient is $\mathrm{tr}(\boldsymbol{G}_i^T \boldsymbol{\Sigma}_i \boldsymbol{G}_i \boldsymbol{\Sigma}_i)$. Vectorizing recovers the familiar $\boldsymbol{g}^T \boldsymbol{\Sigma} \boldsymbol{g}$ form with $\boldsymbol{g} = \mathrm{vec}(\boldsymbol{G})$ and covariance $\boldsymbol{\Sigma} \otimes \boldsymbol{\Sigma}$; we keep the matrix form throughout because the displacement norm of (5) is compact in it. Note that the quadratic form is quartic rather than quadratic in the entrywise noise scale, which is where the $\kappa^{\pm 2}$ exponents of Section 4.3 originate.

A.4 Infrastructure and reproducibility

Because a central claim is a wall-clock reduction, we state the machine: a single NVIDIA A100 GPU with 4 CPU cores and 50 GB of RAM, running CentOS 7.9, Python 3.10.7, PyTorch 2.6.0 (CUDA 12.4), NumPy 2.2.6, Gym 0.26.2 and MuJoCo 2.1.0. Every configuration in a given comparison was timed on the same hardware within a single job, so the ratios are unaffected by scheduling across heterogeneous nodes. Seeds are set through one entry point covering Python, NumPy and PyTorch. The percentile experiment is fully seeded and reproduces bit-for-bit on every non-timing

Table 3: Where the guided return distribution sits relative to the unguided one. Intervals are 95% over 512 paired trajectories. The unguided reference has mean 38.33; its worst-tenth cutoff is 36.31 and the average over that worst tenth, the value the guidance is supposed to reach, is 29.38. The four empirical CDFs coincide to the width of a line (Figure 4).

	paired Δ vs unguided	**quantile**
Unguided (reference)	–	0.434
Ancestral (N=128)	$+0.0025 \pm 0.0020$	0.438
Implicit ($\tilde{N}$=20)	$+0.047 \pm 0.029$	0.451
uncalibrated	$+0.048 \pm 0.028$	0.451

quantity; the earlier sweeps predate that fix and draw fresh conditioning states on each invocation, so their absolute values move by a few tenths of a percent between repetitions while the between-sampler ratios do not, being computed within one invocation on matched batches. The sampler, the pre-registration documents and the logged metric histories will be released with the paper, and are withheld here because links would identify the authors under the double-blind policy.

A.5 Future work

An end-to-end unguided control arm is the one experiment that decides how the percentile result of Section 6.2 should be read, separating the hypothesis that the guidance mechanism works cumulatively from the hypothesis that it contributes nothing while PolyGRAD's world model supplies the performance. Next, pinning the policy-guidance scale instead of tuning it online would turn the destabilization finding of Section 6.4 from an association into a mechanism, since acceleration moving that scale to a five-fold higher operating point is currently the leading but unconfirmed explanation. Beyond that: more seeds and a pre-registered checkpoint inclusion criterion, to move the equivalence tests of Section 6.3 from a single run per configuration to a distribution; and cheap sampler-level sweeps over α and subsequence spacing, weighted toward the noisiest steps, since Section 6.2 shows the adversarial displacement concentrated there. Beyond that, the obstruction of Section 4.3 is worth attacking rather than respecting: calibrating the adversarial step through $\mathrm{tr}(\boldsymbol{G}^T\boldsymbol{\Sigma}\boldsymbol{G}\boldsymbol{\Sigma})$ is what makes the coefficient diverge as the transition becomes deterministic and equally what fixes the displacement at $O(\sigma^2)$ against noise of $O(\sigma)$. A guidance scheme calibrated against something else, for instance the norm of the predicted clean trajectory, would address both at once, which is where we think the next gain lies.

A.6 Experimental detail

This section gives the results of Section 6 in full.

The endpoints that cannot testify. The controls produce a negative result about measurement rather than about the method. Under the uncalibrated arm, whose enforced budget is wrong by $6.2\times$, mean imagined return moves by 0.02; under $\kappa = 10^{-2}$, which switches the adversary off to within a factor of 13 000 (Section 6.1), it moves by less than a quarter of one within-batch standard deviation, in the wrong direction. Dynamics MSE and reward MSE are flat across both controls and the terminal fraction is identically zero. Four statistics conventionally reported for this comparison do not respond when the mechanism they describe is disabled, so none can testify that it is intact. This is why the fidelity results of Section 6.3 are claims about the *world model* and never about the guidance, and it is what motivated the percentile endpoint of Section 6.2.

The shape of the guidance. Along both chains c grows sharply toward the clean end, by 5.3×10^4 for the ancestral arm and 1.5×10^4 for the implicit one (Figure 3). It is the coefficient that grows, not the force, since by (5) the displacement runs the other way; and the two arms are not related by a fixed rescaling of c, the cross-arm ratio at matched noise levels being 0.58 rather than the $\sqrt{N/\tilde{N}} = 2.46$ a naive reading of (4) suggests, because a larger stride enlarges $\boldsymbol{\Sigma}^{\mathcal{S}}_{s_j}$ in the denominator by more than it raises $\log \eta_j$.

The step is too small to have done otherwise. Equation (5) fixes the adversarial step at $\sqrt{2\log\eta_j}\,\sigma_{s_j}^2$, while the noise the same denoising step injects has norm $\sigma_{s_j}\sqrt{d}$, with d the number of entries in a trajectory's observation block ($d = 110$ here: horizon 10, state dimension 11). The adversary is $O(\sigma^2)$ against an $O(\sigma)$ perturbation and $\sigma_{s_j} < 1$ throughout the schedule, so it is dominated at every step: the displacement never exceeds 1.4% of the noise injected beside it and runs nearer 0.1% through the middle of the chain (Figure 3). Summed over all 127 steps it comes to 0.266 against a normalized trajectory of Frobenius norm $\sqrt{d} = 10.5$, so even granting perfect coherence across steps and no re-projection the mechanism has 2.5% of the trajectory's scale to work with, and it does not spend even that: 71.2% is applied in the ten noisiest steps, after which 118 further steps return the sample to the model's manifold. Two things follow. Calibrating the guidance step against the noise *variance* rather than its scale is the shape of the problem, and it comes from (2) rather than from anything acceleration does, which is why the ancestral chain fails the test as squarely as the implicit one. And because (5) makes the magnitude independent of $\boldsymbol{G}$, this cannot be an artifact of the particular return model measured here: a different Z_ϕ changes the direction the adversary pushes, never how far, so the measurement bounds what any direction could achieve at this magnitude.

Two further pre-registered comparisons. The implicit chain tracks the ancestral chain closely ($\Delta q = 0.014$ against a 0.15 threshold), so the acceleration faithfully reproduces the method it accelerates, which is this paper's claim and is unaffected. And the uncalibrated arm, whose enforced budget is wrong by $6.2\times$, is indistinguishable from the calibrated one ($\Delta q = 0.000$); we draw the conservative reading of that. The recalibration restores a guarantee the uncorrected sampler misstates, and at $\alpha = 0.1$ it has no detectable consequence for any distributional endpoint we can measure, because the mechanism whose strength it sets is itself below detection: a budget error and a correct budget are equally invisible when the quantity being budgeted cannot move the outcome. The recalibration should be adopted for the reason a correct implementation is adopted, not on the strength of a measured improvement, which we do not have.

What this does and does not establish. It is a sampler-level measurement on one trained checkpoint, and the guidance acts cumulatively across $375\,000$ policy updates during training, so it does not show that adversarial diffusion fails end to end; the robustness results of Foffano et al. (2026) are not addressed by it. What it does show is that a single batch of imagination is not detectably steered, which bounds what any sampler-level experiment can establish about the risk mechanism, and which explains why the fidelity statistics of Section 6.3 are flat across every control we ran.

An asymmetry worth naming. Fidelity does not replicate nearly as tightly. The quantity tested is a per-checkpoint ratio: each checkpoint's terminal dynamics MSE under the implicit sampler divided by its own ancestral value, both means over five batches. On the reference checkpoint of Table 1 this gives $0.1241/0.1230 = 1.009$, 1.046 and 1.043 at $\tilde{N} = 20, 15, 10$; the group figures that follow are means of these per-checkpoint ratios over the four checkpoints in each seed-paired group of *further* checkpoints, and do not include the reference checkpoint. Against an equivalence margin of 1.10 on this ratio, fixed before any of these data were collected (Schuirmann, 1987; Lakens, 2017), the first group declares all three implicit samplers equivalent to the ancestral chain (ratios 0.997, 1.044, 1.045); in the second, $\tilde{N} = 20$ is equivalent (ratio 1.000) while $\tilde{N} = 15$ and 10 return *inconclusive* verdicts (ratios 1.021, 1.011), the only two configurations in the study not declared equivalent (Figure 5). Their point estimates sit at parity and their intervals are widened by one checkpoint whose world model is markedly noisier. We report these as inconclusive rather than as failures, because the margin was fixed in advance and the intervals do not clear it, and we record that this is a power failure rather than a negative result: the between-checkpoint standard deviation is 0.049 across the homogeneous checkpoints but 0.167 with the degenerate one included, and a pre-registered inclusion criterion would have brought the required n from twelve to six. We did not have one, and a replication should. Across the nine checkpoints reward error tells a cleaner story than dynamics error: it is consistently about 1.6% higher for the implicit sampler, worse in 9 of 9 checkpoints at $\tilde{N} = 20$, a sign test at $p \approx 0.004$, so the effect is systematic and real while also being negligible in size and no worse at ten steps than at twenty. Dynamics error is indistinguishable in aggregate: pooling the reference checkpoint with both groups, the mean per-checkpoint ratio over all nine is 1.00, 1.03, 1.03, with checkpoint-to-checkpoint spread running in both directions. None of these columns is evidence about the guidance, which is flat when the adversary is switched off

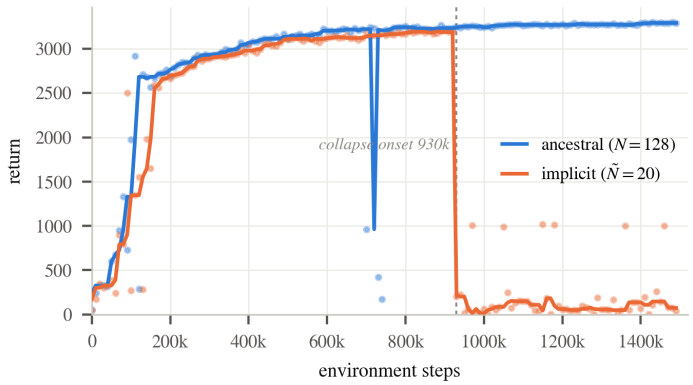


Figure 2: Training with the sampler in the loop, matched seed and matched environment steps, both arms complete to 1.5M; points are logged evaluations, lines a rolling median. The accelerated arm reaches parity by 860k (3220 against 3305) at a fifth of the wall-clock, then collapses at 930k and does not recover. The ancestral arm's dips at 700–740k are transient.

Table 4: The $K = 1$ matrix: three environments, both arms, seed 1, 1.5M steps. "tail" is the mean of the last ten evaluations. The Hopper collapse reproduces at a different ensemble size, but on Walker2d it is the *ancestral* arm that is unstable, so the failure is not a property of the sampler. †HalfCheetah's ancestral arm reached only 970k steps and is not comparable at equal steps.

Env	Arm	peak	tail	tail/peak	outcome
Hopper	ancestral	3346	3333	0.996	stable
Hopper	**implicit**	3123	**97**	**0.031**	collapsed
Walker2d	**ancestral**	4035	**1835**	**0.455**	unstable
Walker2d	implicit	4163	3957	0.951	stable
HalfCheetah	ancestral	4685	4599	0.982	incomplete†
HalfCheetah	implicit	4330	3978	0.919	stable

(Section 6.1); what they establish is fidelity of the *world model* under acceleration, and it is the claim the equivalence test is powered for.

What breaks is not the sampler. The obvious explanation, accumulated fidelity loss from the shorter chain, is the one the data rule out. Across all 128 logged metrics scored for movement in the 70k steps before onset, the largest excess on any pre-registered hypothesis is $|z| = 1.20$, for world-model fidelity, and the accelerated arm's imagination error runs 3–5× *lower* than the baseline's throughout. What does differ is a control loop in the surrounding algorithm: a policy-guidance scale that AD-RRL inherits from PolyGRAD and tunes online, which the accelerated arm holds at 5.1× the baseline's operating point and which becomes slaved to its own tracking error once the collapse begins ($\rho = 0.983$ against 0.434). Appendix F gives the screen, the loop analysis, the robustness sweep, and a second paired run on Walker2d at $K = 5$ that degrades gradually rather than breaking. Two cautions carry into the body: the loop is indistinguishable between arms *before* onset, so this run cannot separate a loop that initiates the collapse from one that merely amplifies something else; and what first perturbs the system at ∼929k is unidentified.

The deterministic limit. Section 4.3 predicts $c = O(\kappa^{-2})$ and a displacement $O(\kappa^2)$. Sweeping κ from 1 to 10^{-2} at $\tilde{N} = 20$ gives fitted log-log slopes of -2.00 and $+2.00$, and $\kappa = 0$ raises the singularity in (4) directly. Being exact consequences of (5) rather than noisy estimates, these verify that the implementation realizes the stated construction; the substantive content is the failure mode. At $\kappa = 10^{-2}$ the adversarial displacement is 1.3×10^{-6} against 1.3×10^{-2} at $\kappa = 1$, a factor of 13 000, while the sampler goes on producing ordinary-looking trajectories. A deterministic solver does not announce that it has disabled the adversary; it generates unguided samples under a guided interface, and that silence, rather than the undefined limit, is the practical reason the obstruction of Section 4.3 matters.

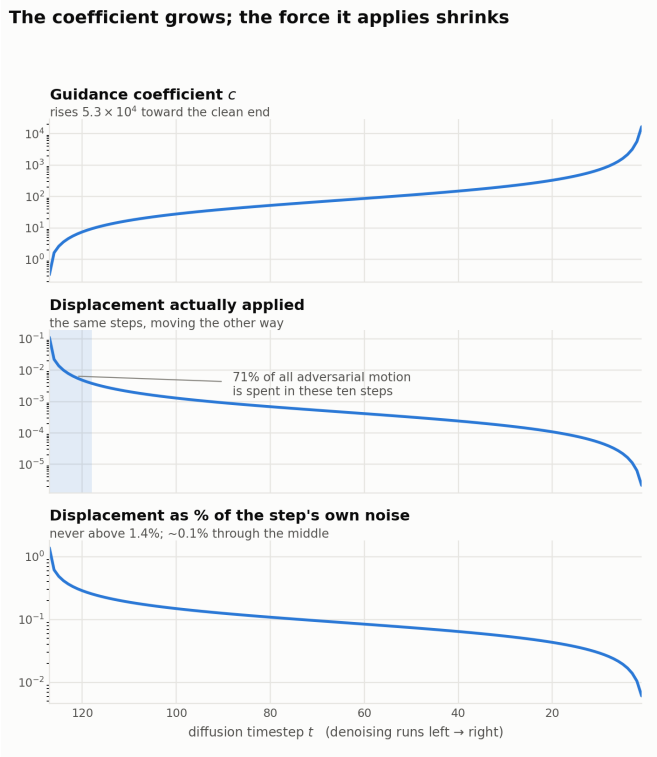


Figure 3: Why the guidance cannot move the distribution. **Top:** the coefficient c climbs toward the clean end of the chain. **Middle:** the displacement it actually applies falls across the same steps, so the two run in opposite directions and c must not be read as guidance strength. **Bottom:** that displacement as a percentage of the noise the same step injects. These are derivations rather than measurements: every curve follows from the noise schedule and (5) alone, with no checkpoint involved, which is why the conclusion does not depend on the trained model.

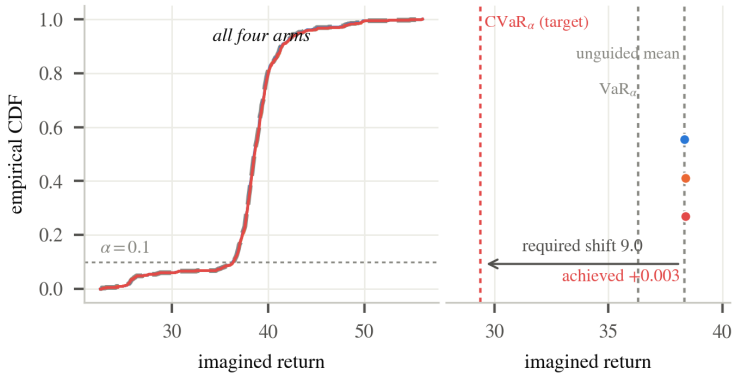


Figure 4: **Left:** empirical CDFs of imagined return for all four conditions; they coincide. **Right:** where each guided arm's mean landed. CVaR_α marks where the risk objective says the guided distribution should sit; every arm sits at the unguided mean instead.

A.7 Additional figures

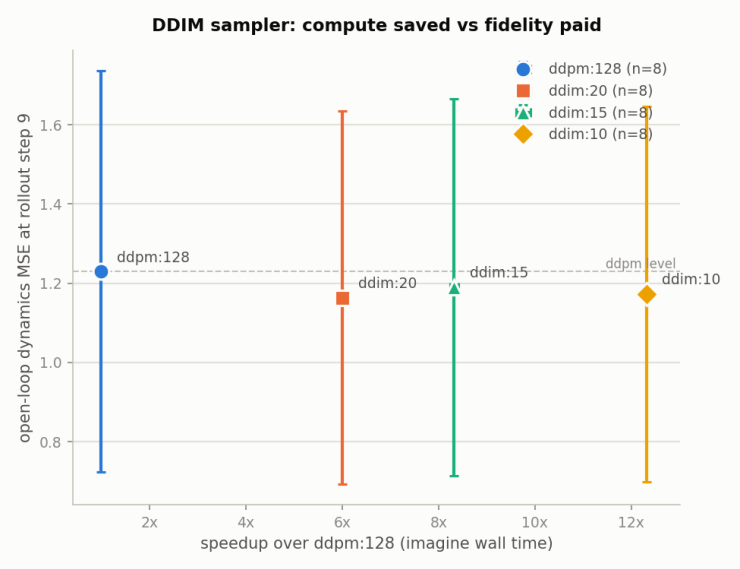


Figure 5: Compute saved against fidelity paid, over the nine independently trained Hopper checkpoints. Vertical axis: open-loop dynamics MSE at the final rollout step. Horizontal axis: measured speedup relative to the full ancestral chain. Bars are 95% confidence intervals across checkpoints; the dashed line marks the ancestral level. Open markers denote the two configurations whose equivalence verdict was inconclusive.

B IMPLICIT SAMPLING: THE FULL EXPRESSIONS

Section 4.1 states the implicit reverse step in words and gives only its variance. Here it is in full. Write $\bar{\alpha}_i$ for the cumulative noise level at step i, $\epsilon_{\boldsymbol{\theta}}$ for the network's noise prediction, and $\hat{\boldsymbol{\tau}}_0$ for the clean-trajectory prediction it induces. For a subsequence $\mathcal{S} = \{s_1 < \cdots < s_{\tilde{N}}\}$ with $s_1 = 0$, initialized at $\boldsymbol{\tau}_{s_{\tilde{N}}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, the reverse step is

$$p_{\boldsymbol{\theta}}(\boldsymbol{\tau}_{s_{j-1}} | \boldsymbol{\tau}_{s_j}) = \mathcal{N}\left(\mu^{\mathcal{S}}_{\boldsymbol{\theta}}(\boldsymbol{\tau}_{s_j}, s_j), \boldsymbol{\Sigma}^{\mathcal{S}}_{s_j}\right), \tag{6}$$

with

$$\mu^{\mathcal{S}}_{\boldsymbol{\theta}}(\boldsymbol{\tau}_{s_j}, s_j) = \sqrt{\bar{\alpha}_{s_{j-1}}}\ \hat{\boldsymbol{\tau}}_0(\boldsymbol{\tau}_{s_j}) + \sqrt{1 - \bar{\alpha}_{s_{j-1}} - \sigma^2_{s_j}}\ \epsilon_{\boldsymbol{\theta}}(\boldsymbol{\tau}_{s_j}, s_j), \tag{7}$$

and the schedule factor ρ_j of (3) given by

$$\rho_j = \sqrt{\frac{1-\bar{\alpha}_{s_{j-1}}}{1-\bar{\alpha}_{s_j}}}\sqrt{1 - \frac{\bar{\alpha}_{s_j}}{\bar{\alpha}_{s_{j-1}}}}. \tag{8}$$

The initialization is valid whenever the largest retained step carries essentially all of the forward noise. Because (6) is Gaussian with an explicit mean and covariance, the adversarial guidance of (1) applies to it unchanged:

$$\bar{p}_{\boldsymbol{\theta}}(\boldsymbol{\tau}_{s_{j-1}} \mid \boldsymbol{\tau}_{s_j}) = \mathcal{N}\big(\mu^{\mathcal{S}}_{\boldsymbol{\theta}}(\boldsymbol{\tau}_{s_j}, s_j) - c_{s_j}\boldsymbol{\Sigma}^{\mathcal{S}}_{s_j}\boldsymbol{G}_{s_j}\boldsymbol{\Sigma}^{\mathcal{S}}_{s_j},\ \boldsymbol{\Sigma}^{\mathcal{S}}_{s_j}\big). \tag{9}$$

C DERIVATIONS

This appendix supplies the algebra behind the two claims of Section 4 that the body states in prose: the chain-length recalibration of Section 4.2, and the displacement norm (5) from which both the deterministic-limit obstruction of Section 4.3 and the fixed-norm property of Section 4.4 follow.

Chain-length recalibration. Let n be the number of steps of $\mathcal{S}$ that carry guidance. Applying n factors of $(1/\alpha)^{1/N}$ yields $(1/\alpha)^{n/N}$, which equals $1/\alpha$ only when $n = N$. Substituting $\eta_j(\alpha, n) = (1/\alpha)^{1/n}$ gives $\prod_{j=1}^{n} \eta_j = 1/\alpha$ by construction. Calibrating instead against the realized subsequence length $\tilde{N} > n$, as the sampler used for our experiments does, yields $(1/\alpha)^{n/\tilde{N}}$, a shortfall of $(1/\alpha)^{1/\tilde{N}}$, that is, one step's share. For the second claim, the argument of Foffano et al. (2026) requires only that the trajectories lie in a bounded set and that the transition covariance be diagonal with entries in $[0, 1)$. By (3) the implicit covariance factor $\sigma^2_{s_j}\boldsymbol{I}$ is diagonal, and for $\kappa \leq 1$ its entries lie in $[0, 1)$ wherever the ancestral schedule's do. The derivation is otherwise unchanged under $i \mapsto s_j$ and $N \mapsto n$.

The displacement norm, the deterministic limit, and the fixed-norm step. By Appendix A.3 the reverse transition is matrix-normal with left and right covariance factors $\boldsymbol{U} = \boldsymbol{V} = \boldsymbol{\Sigma}^{\mathcal{S}}_{s_j} = \sigma^2_{s_j}\boldsymbol{I}$, so the guidance drift is $\boldsymbol{UGV} = \sigma^4_{s_j}\boldsymbol{G}$ and the normalizing quadratic form in (4) is $\mathrm{tr}(\boldsymbol{G}^T\boldsymbol{UGV}) = \sigma^4_{s_j}\|\boldsymbol{G}\|^2_F$. Then

$$c_{s_j} = \frac{\sqrt{2\log\eta_j}}{\sigma^2_{s_j}\|\boldsymbol{G}\|_F}, \qquad \big\|c_{s_j}\boldsymbol{UGV}\big\|_F = \sqrt{2\log\eta_j}\ \sigma^2_{s_j}, \tag{10}$$

which is (5). The right-hand expression contains no $\boldsymbol{G}$, because $\|\boldsymbol{G}\|_F$ cancels between the two factors, which is the fixed-norm property of Section 4.4. By (3) $\sigma_{s_j} \propto \kappa$, so the coefficient grows as κ^{-2} and the displacement vanishes as κ^2, which is the obstruction of Section 4.3. The quartic dependence on σ_{s_j} in the quadratic form, and hence the exponent 2 rather than 1, comes from the covariance appearing on both sides of $\boldsymbol{G}$; a vector-valued construction with a single covariance factor would give $\kappa^{\mp 1}$ instead.

D VERIFICATION: IMPLEMENTATION MATCHES THEORY

Several quantities the sampler logs are algebraically determined rather than measured. Reporting them among the results would imply empirical support they cannot give, so we report them here as evidence that the implementation realizes the construction of Section 4.

The budget product. On an implementation calibrated to the guided-step count the realized product $\alpha \prod_j \eta_j$ equals one by construction, and no experimental outcome could refute it. The middle column below is the sampler as run, which calibrates against the realized subsequence length $\tilde{N}$; the right column is what Section 4.2 prescribes.

	guided steps	$\alpha \prod_j \eta_j$ (as run)	corrected
Ancestral (N=128)	127	0.982	1.000
Implicit ($\tilde{N}$=20)	21	0.901	1.000
uncalibrated	21	0.146	0.146
Implicit ($\tilde{N}$=15)	15	0.866	1.000
uncalibrated	15	0.131	0.131
Implicit ($\tilde{N}$=10)	10	0.811	1.000
uncalibrated	10	0.120	0.120

The as-run rows fall short of one for the reason given in Section 4.2: only $n = \tilde{N} - 1$ shares are ever spent against a denominator of $\tilde{N}$. Correcting this multiplies every coefficient by $\sqrt{\tilde{N}/n}$, which is 1.024, 1.033 and 1.049 at $\tilde{N} = 20$, 15 and 10. The uncalibrated rows carry full-chain shares by construction, so the correction does not apply to them, and the gap between them and the recalibrated rows is the comparison reported in Section 6.1.

The displacement norm. By (5) the per-step displacement is $\sqrt{2 \log \eta_j}\, \sigma_{s_j}^2$ exactly. Reconstructing it from the noise schedule alone and comparing against the values the sampler logs therefore tests the implementation, not the theory. The two agree: predicted maximum 0.111009 against logged 0.111009, predicted chain mean 0.002077 against logged 0.002093, predicted sum over the chain 0.2658 against logged 0.266.

The κ exponents. The sweep of Section 6.1 is the same kind of check in a different variable, and we read its fitted slopes the same way: they confirm that the code realizes (5); the substantive content of the sweep is the failure mode rather than the exponents.

E Implementation details

Our implementation extends the public AD-RRL codebase of Foffano et al. (2026), which builds on PolyGRAD (Rigter et al., 2024). The architecture, training procedure and hyperparameters recorded here are AD-RRL's, reproduced so that our sampler comparison is interpretable; the sampler (Algorithm 1), the budget recalibration (4) and the budget instrumentation are ours. Three MLPs are involved: the policy $\pi_{\boldsymbol{\omega}}$, the adversarial denoising diffusion model $\bar{p}_{\boldsymbol{\theta}}$ and the learned cumulative reward model $Z_{\boldsymbol{\phi}}$.

Policy network and training. The policy is parameterized as in PolyGRAD (Rigter et al., 2024): a Gaussian $\pi_{\boldsymbol{\omega}} = \mathcal{N}(\mu_{\boldsymbol{\omega}}(s), \sigma_{\boldsymbol{\omega}})$ with a single state-independent learnable standard deviation, trained with A2C and Generalised Advantage Estimation using ADAM (Table 5).

Diffusion and reward models. We follow PolyGRAD's training procedure and MLP architecture for the diffusion model, minimizing an L2 loss with ADAM, with skip connections at every layer and a learnable embedding of the diffusion step i (Janner et al., 2022) (Table 5). The cumulative reward model $Z_{\boldsymbol{\phi}}$ shares that structure and step embedding, with a final linear layer producing the scalar return. Training alternates the two: at each iteration $\bar{p}_{\boldsymbol{\theta}}$ is improved on the usual denoising objective (Ho et al., 2020) over trajectories drawn from the buffer, and $Z_{\boldsymbol{\phi}}$ is fit to predict $Z(\boldsymbol{\tau}_0)$.

Columns omitted from Table 1. Mean imagined return is 38.76 for the ancestral chain and 38.80 for all three implicit samplers, agreeing to under 0.11%. First-step reward MSE differs by under 1% across rows. Neither responds when the adversarial guidance is disabled (Section 6.1), so neither is evidence about the guidance.

Table 5: Hyperparameters for A2C training (left) and for adversarial diffusion training (right).

A2C	Value	Diffusion	Value
Batch size	512	Hidden size	1024
Synthetic traj. length	10	Generated traj. length	10
GAE λ	0.9	Batch size	256
Critic learning rate	3e-4	Step embedding size	128
Actor learning rate	3e-5	Number of layers	6
Discount factor, γ	0.99	Learning rate	3e-4
Entropy bonus weight	1e-5		

Reward-model ensemble. The reward model is an ensemble of K heads over a shared trunk, aggregated into a single estimate before the guidance gradient is taken; $K = 1$ recovers the single-estimator formulation of Foffano et al. (2026). The experiments of Section 6 use $K = 5$ with mean aggregation and a per-head bootstrap keep-probability of 0.8, which is what our reference checkpoints were trained under. We record this explicitly because it is a deviation from the method we accelerate: K was inherited from a configuration default rather than chosen, and we found it only on auditing the saved weights. It does not affect the sampler comparison, since K enters only through the guidance direction $\boldsymbol{G}$ and not through the denoising chain, the transition covariance, the budget product, or, by (5), the displacement magnitude. The nine checkpoints of Section 6.3 are consequently a mix: one reference checkpoint and four ablation checkpoints at $K = 5$, four at $K = 1$. Both arms of the original pair in Section 6.4 use $K = 5$, so that pairing is matched; the replication matrix in the same subsection is run at $K = 1$ throughout, which is what makes it a test of the earlier result rather than a repetition of it.

F END-TO-END FORENSICS

This appendix carries the pre-onset metric screen, the control-loop analysis, the robustness sweep, and the second paired run on Walker2d at $K = 5$.

The onset screen. Each of the 128 logged metrics is scored for movement in the 70k steps before onset as a robust z against its own earlier baseline, then differenced against the same measurement on the stable arm. The largest excess on any pre-registered hypothesis is $|z| = 1.20$ for world-model fidelity, 0.83 for value overestimation, 0.57 for trust-region pathology and 0.37 for the guidance budget. Nothing gives warning, and two hypotheses are positively excluded rather than merely unsupported: across the onset the imagined and environment value estimates track to within 1%, and there is neither a KL spike nor line-search saturation. The accelerated arm is not a degraded one on any fidelity metric we log, its imagination error running 3–5$\times$ *lower* than the baseline's throughout.

The control loop. Separately from the CVaR coefficient c_i, AD-RRL inherits from PolyGRAD a scalar *policy-guidance scale* that keeps imagined actions consistent with the policy, and tunes it online through a gradient step on its logarithm proportional to $(\mathrm{act_std} - 1)$, holding the imagined action spread at unit scale. The update raises the scale when the spread is too high, and more guidance produces more displacement, which raises the spread further, so the loop is positive feedback by construction and is quiescent only while the spread stays inside a narrow band. Before onset the two arms' loops are statistically indistinguishable: mean $|\mathrm{act_std} - 1|$ of 0.0054 against 0.0048, and correlation between the tracking error and the next update of 0.273 against 0.274. After onset the accelerated arm's error grows $8.5\times$ in mean and $12\times$ in maximum, and the scale becomes almost perfectly slaved to it ($\rho = 0.983$), while the control arm barely moves (0.0053, $\rho = 0.434$).

The loop is where the failure becomes visible, and it is why the collapse is permanent rather than transient: once the scale is slaved to its own tracking error it drives the system further from the band it is meant to restore, whereas the ancestral arm's comparable dips resolve within one or two evaluations. But the two loops are indistinguishable *before* onset, so this run cannot separate a loop that initiates the collapse from one that only amplifies a disturbance arriving from elsewhere. Deciding between those requires intervening on the controller, which we have not done here.

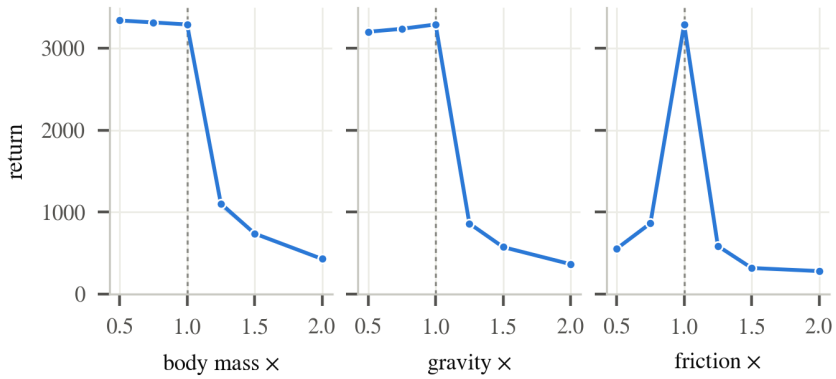


Figure 6: Test-time robustness of both trained policies at the step-500k checkpoint, where both are intact and their nominal returns agree to 0.2% (3104 against 3109). Median absolute difference across the 18 conditions is 1.1%. Bands are minimum to maximum over the 20 evaluation episodes at each setting and reflect the initial-state distribution only, not seed uncertainty, which is far larger on this task and which one paired seed cannot express. The wide ancestral band at gravity $\times 0.5$ is a bimodal cell (413 to 3076 over 20 episodes). This comparison is descriptive.

Where the elevated operating point comes from. The standing difference that makes this arm the one to run away is its operating point: through the whole stable phase it holds a guidance scale $5.1\times$ the baseline's, and the Walker2d pair below replicates the elevation at $5.62\times$. Neither arm was tuned for this, since both inherit the same initial value and the same learning rate and the loop finds its own level. One arithmetic observation is worth recording. The loop holds a quantity accumulated across guided steps at a fixed target, and the two arms apply 127 and 21 guided steps, a ratio of 6.05. A per-step scale inversely proportional to the step count would settle at $6.05\times$ the baseline's, against $5.1\times$ and $5.62\times$ observed, the shortfall being what sub-linear accumulation would produce. We offer this as a consistency check on the magnitude rather than a derivation, because the policy-guidance mechanism is PolyGRAD's and is not the CVaR construction analysed in Section 4. If it is the right reading, the elevation is a generic consequence of shortening a chain that a feedback-tuned scale is wrapped around, and pinning the scale tests it. The off-by-one budget denominator of Section 4.2 biases short chains the same way, but its $\sim 2\%$ coefficient error is the right sign and the wrong order of magnitude to account for a factor of five alone.